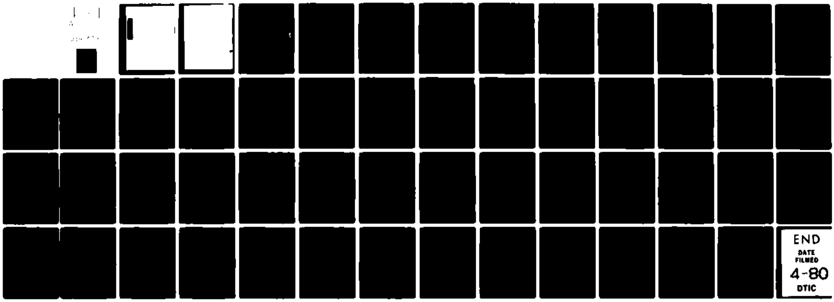
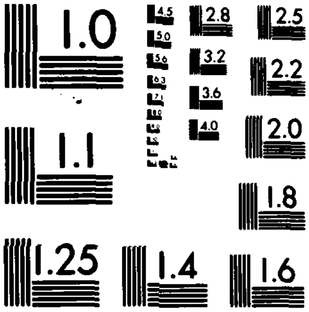
AD-A081 779 NAVAL SURFACE WEAPONS CENTER DAHLGREN LAB VA F/G 12/1
MATRIX ARITHMETIC AND CHARACTERISTICS COMPUTATION,(U)
OCT 77 A V HERSHEY
UNCLASSIFIED NSWC/DL-TR-3689 NL



END
DATE
FILMED
4-80
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ADA081779

8

14 NSWC/DL-TR-3689

11 Oct~~ober 1~~977

6 MATRIX ARITHMETIC
AND
CHARACTERISTICS COMPUTATION

DTIC
ELECTE
MAR 1 2 1980
S
C
D

By

10 A. V. HERSHEY

Science and Mathematics Research Group

12 53

Approved for public release; distribution unlimited.

391 597

TABLE OF CONTENTS

	Page
TABLE OF CONTENTS	i
FOREWORD	ii
ABSTRACT	iii
INTRODUCTION	1
MATRIX SPECIFICATIONS	3
MATRIX NOTATION	4
MATRIX ARITHMETIC	5
Analysis	5
Programming	9
PARTITIONED MATRIX ARITHMETIC	10
Analysis	10
Programming	11
CHARACTERISTIC ROOTS AND VECTORS	13
Analysis	13
Programming	18
DISCUSSION	19
CONCLUSION	20
BIBLIOGRAPHY	21
APPENDICES	
A. MATRIX INVERSION	
Conjugate Gradient	
B. CHARACTERISTIC ROOTS AND VECTORS	
Companion Matrix	
Reduction to Companion	
Tridiagonalization	
Minimized Iteration	
C. TEST MATRICES	
D. DISTRIBUTION	

Accession For
NTIS GRA&I ☑
DDC TAB □
Unannounced □
Justification______
By______
Distribution/
Availability Codes
Dist | Avail and/or special
A

FOREWORD

Investigations of matrix operations have been made in this laboratory for many years. Subroutines have been prepared on the basis of many methods of computation. Before experience with the subroutines could be reported, the subroutines became obsolete as the result of replacements of the computers. It is the purpose of this report to document a few surviving subroutines. The manuscript was completed by 27 October 1977.

Released by:
Ralph A. Niemann
Ralph A. Niemann
Head, Strategic Systems Department

ABSTRACT

Analysis and documentation are given for subroutines which do matrix arithmetic and characteristics computation. The subroutines make it possible for the elements of the matrix to be in the natural arrangement. During inversion or trianguloidization the pivot selection is independent of the scaling of the rows and columns. A subroutine does arithmetic on partitioned matrices.

INTRODUCTION

In an early application, the Danilevski method was used to compute the characteristic roots of a large economic matrix on the Mark II Aiken Relay Calculator for Professor Morgenstern of Princeton University. Experiments on matrix operations were continued on The Naval Ordnance Research Calculator. An early library of subroutines was based on a variety of algorithms which had been extolled in the literature. Matrix routines which were prepared for NORC became useless with the dismantlement of NORC. They were salvaged by a conversion to a hybrid version of FORTRAN which would run on STRETCH. Even the hybrid version of FORTRAN became useless with the dismantlement of STRETCH. Now the computer in this laboratory is a CDC 6700 Computer.

In the meantime there have been giant projects by large staffs of personnel for the preparation of subroutines to perform all operations on many forms of matrix. There are the Math Sciences Library of Control Data Corporation[1], the System/360 Scientific Subroutine Package of the International Business Machines Corporation[2], and the Subroutine Library of the International Mathematical and Statistical Libraries, Inc.[3] Subroutines for the computation of roots and vectors are available in a Package of Matrix Eigensystem Routines[4]. Nevertheless, some of the ideas in the original NORC library still are worth recognition and are documented in the present report.

Available methods for the transference, transposition, addition, subtraction, or multiplication of matrices are standard[5]. Available methods for the inversion of matrices or for the determination of characteristic roots and vectors are numerous. The merits of many methods have been discussed by Wilkinson[10,12]. The methods have fluctuated in popularity. When a method was strongly defended it was programmed for NORC. Experience on the computer has guided the selection of methods for the present investigation. Under consideration are only those methods which are suitable for the general matrix with arbitrary symmetry. Not included are the special methods which are useful only for sparse or symmetric matrices.

Matrix inversion by the evaluation of determinants is inefficient for large matrices because of redundancy. Reduction of a matrix to the coefficients of a characteristic polynomial fails because a polynomial of high degree cannot be evaluated near its largest roots.

The application of sequential methods to asymmetric matrices can get into trouble from vanishing pivots. Matrix inversion by the reduction of a matrix to diagonal form through progressive partitioning fails if the matrix has a zero principal minor determinant. Determination of characteristic roots by the reduction of a matrix to tridiagonal form through progressive partitioning fails if the sum of products of elements in a row and a column is zero.

Biorthogonalization methods have a strong appeal from a philosophical standpoint because they work with vector invariants. Unfortunately, whenever two vector invariants of different magnitudes are combined into a single vector, the accuracy of the smaller vector is lost to round off. Also, the biorthogonalization in a three–term recurrence deteriorates as the recurrence advances, and the deterioration must be kept in check with a multi–term recurrence.

Iterative methods are the most accurate insofar as they refer in each cycle to the original matrix. In too many cases the rate of convergence of the iteration is too slow, and the convergence must be accelerated by various techniques. Also, it is necessary to make a suitable choice of initial conditions for the iteration.

The popularity of elemental methods has run full circle. In the beginning the favorite method was Gauss elimination with pivot selection. Now the favorite method again has become the Gauss elimination with pivot selection.

The best method for the determination of the characteristic roots and vectors reduces the matrix to a tractable form with the least amount of damage. A symmetric matrix can be reduced to tridiagonal form by the Householder reduction. An asymmetric matrix can be reduced to trianguloid form by permutations and eliminations. The roots of a triangular matrix are just the diagonal elements. Similarity transformations are applied in the QR algorithm to bring the matrix to blockwise triangular form. In the present investigation the roots of the trianguloid matrix are derived by Newton–Raphson iteration.

Although there exists already a subroutine MATRIX in the CDC 6700 system, a new subroutine MTRX has been prepared. The new subroutine allows the interval between columns to be other than +1, and permits the matrices to have their natural arrangement instead of the transposed arrangement. The subroutine MATRIX is written in machine language whereas the subroutine MTRX is written in FORTRAN. Timing trials have shown that the speed of MTRX could be increased if the subroutine were rewritten in machine language. This would not be true if the FORTRAN compiler were optimum.

The NORC was a three–address binary coded decimal computer. The normal function of a three–address instruction was arithmetic on numbers in core storage. If the three–address instruction were preceded by a call to a matrix subroutine, then the three addresses were reinterpreted as specifications for whole matrices. The coding for matrix arithmetic was little more difficult than the coding for number arithmetic.

When matrices were too large to fit into the core memory of NORC, they were partitioned and were stored on tape. The matrices could be partitioned rowwise or blockwise and they could be prepositioned or repositioned before operation. The computing loops were designed for minimum time of operation and the tape movements were designed for minimum time of repositioning. It took six months to design the partitioned matrix routine for NORC. Translation of the original routine to FORTRAN would not be feasible because many capabilities which were under program control on NORC are no longer under program control in FORTRAN.

On NORC it was possible to process addresses in core with meta arithmetic. In FORTRAN there is no program control over addresses except through indices.

On NORC a two–dimensional array could be arranged in the natural order of mathematics. In FORTRAN the two–dimensional array is stored in transposed order in memory.

On NORC all data were numeric and only four of the seven tracks were used on tape. Seven tape units were available to the programmer. Each tape was designated in the program by a symbolic unit number. In FORTRAN the data are alphanumeric or binary. BCD data are written in even parity while binary data are written in odd parity. Six tape units and fifty disk files are available to the programmer. Disk files are designated by the same symbolic unit numbers as tape files. Which device actually is associated with each unit number is specified by a device definition statement.

On NORC each block was given a block number. Blocks could be read forward or backward. Intermediate blocks were skipped during the search for a specified block. In FORTRAN there is no provision for record numbers and records must be read forward or backspaced one at a time.

On NORC the writing of interrecord gaps was under program control. By the use of specially designed gaps between sentinel blocks and matrix blocks it was possible to rewrite a matrix anywhere in the interior of a file of matrices. In FORTRAN the record gap is not under program control and a matrix within a tape file can be modified only while the entire file is copied onto another tape file.

On NORC the writing of an EOF was under program control. In FORTRAN an EOF is written automatically whenever a rewind or a backspace is preceded by a write. A rewind instruction to a tape unit causes delay while the tape is rewound physically, whereas a rewind instruction to a disk unit merely resets an index.

For operations with partitioned matrices it is necessary to adopt limitations in order to keep the number of options to a manageable level. Only conversions of format are applied to matrices in BCD format. Transfers of data from or to matrices in BCD format are by read and write instructions. Any number of BCD matrices may be located on each tape file. Matrix operations are performed only on matrices in binary format. Transfers of data from or to matrices in binary format are by buffer in and buffer out instructions. Only one binary matrix is assigned to each tape file.

On the CDC 6700 Computer the maximum order of matrix in core under nominal conditions would be 144 for multiplication and 256 for inversion. The manipulation of larger matrices requires that the matrices be partitioned with the submatrices stored on disk. The size of matrices is limited by the availability of disk storage. The default value for the maximum number of words in disk storage per setup is 896 × 4096 or 3670016. The limit can be increased fourfold with the aid of a limit card to a maximum of 14680064. As long as the maximum number of words is not exceeded, matrices may be reread or rewritten indefinitely on disk.

MATRIX SPECIFICATIONS

The elements of an $m \times n$ matrix A have the arrangement

$$a_{11}, \cdots, a_{1n}, \cdots, a_{m1}, \cdots, a_{mn}$$

because then they are processed in serial order during a matrix–vector multiplication. The elements are stored in the array AA. The matrix may have a spacing l between columns and a spacing k between rows. The structure of the matrix is specified in core storage by the words in an array MA such that

MA(1) = l = interval between columns.

MA(2) = m = number of rows.

MA(3) = n = number of columns.

MA(4) = k = interval between rows.

The structure of the matrix is specified by the four numbers

$$l, \quad m, \quad n, \quad k$$

A compact vector of length n may be specified either as a row matrix with the specification

$$1, \quad 1, \quad n, \quad n$$

or as a column matrix with the specification

$$1, \quad n, \quad 1, \quad 1$$

The product of a row matrix and a column matrix is a scalar product between two vectors while the product of a column matrix and a row matrix is the dyadic product of two vectors. A compact matrix of size $m \times n$ would have the specification

$$1, \qquad m, \qquad n, \qquad n$$

while the transpose would have the specification

$$n, \qquad n, \qquad m, \qquad 1$$

for the same actual arrangement. A submatrix in a larger matrix of order N would have the specification

$$1, \qquad m, \qquad n, \qquad N$$

A null matrix with the specification

$$1, \qquad n, \qquad n, \qquad n$$

can be created by transfer from a matrix which has the specification

$$0, \qquad n, \qquad n, \qquad 0$$

and has a single element equal to zero. The null matrix is converted into the identity matrix when the diagonal elements with the specification

$$1, \qquad n, \qquad 1, \qquad n+1$$

are created by transfer from a matrix with a single element equal to unity. The real parts and the imaginary parts of a matrix of complex numbers would be matrices with the specification

$$2, \qquad m, \qquad n, \qquad 2n$$

All matrices which are involved in a matrix operation must have compatible dimensions.

All matrices on tape files are compact. The structure of a matrix is specified on tape files by a sentinel block which precedes the matrix block. The sentinel block specifies the dimensions m, n of the matrix. In the case of a BCD matrix the dimensions are punched on a single card which precedes the cards of the matrix. In the case of a binary matrix the dimensions are written in a 2-word record which precedes the mn-word record of the matrix.

MATRIX NOTATION

For definitions and theorems, reference may be made to many texts on matrix algebra. An $m \times n$ matrix A is an ordered rectangular array of numbers a_{ij} with the following arrangement

$$A = \begin{Vmatrix} a_{11} & \cdots & a_{1j} & \cdots & a_{1n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{i1} & \cdots & a_{ij} & \cdots & a_{in} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m1} & \cdots & a_{mj} & \cdots & a_{mn} \end{Vmatrix} \tag{1}$$

Matrix A may be transposed by interchange of rows and columns into matrix A'. Every nonsingular square matrix A has an inverse matrix A^{-1} such that

$$A \cdot A^{-1} = I \qquad (2)$$

where I is the identity matrix. The inverse matrix is given in accordance with Cramer's rule by the equation

$$A^{-1} = \left\| \begin{matrix} \frac{A_{11}}{|A|} & \cdots & \frac{A_{1j}}{|A|} & \cdots & \frac{A_{1n}}{|A|} \\ \frac{A_{i1}}{|A|} & \cdots & \frac{A_{ij}}{|A|} & \cdots & \frac{A_{in}}{|A|} \\ \frac{A_{n1}}{|A|} & \cdots & \frac{A_{nj}}{|A|} & \cdots & \frac{A_{nn}}{|A|} \end{matrix} \right\| \qquad (3)$$

where the determinant A_{ij} is the cofactor of the element a_{ji} in the determinant |A|. The Cramer's rule is the rule to use for matrices up to the third order, but it requires too much redundant computation for matrices of larger order.

MATRIX ARITHMETIC

Analysis

Operations

When matrix A is transferred to matrix B the elements are related by the transformation

$$b_{ij} \rightarrow a_{ij} \qquad (1 \leqq i \leqq m,\ 1 \leqq j \leqq n) \quad (4)$$

When matrix A is transposed into matrix B the elements are related by the transformation

$$b_{ij} \rightarrow a_{ji} \qquad (1 \leqq i \leqq n,\ 1 \leqq j \leqq m) \quad (5)$$

If the matrix C is the sum A + B of the matrices A, B, the elements are related by the transformation

$$c_{ij} \rightarrow a_{ij} + b_{ij} \qquad (1 \leqq i \leqq m,\ 1 \leqq j \leqq n) \quad (6)$$

and if the matrix C is the difference A − B of the matrices A, B, the elements are related by the transformation

$$c_{ij} \rightarrow a_{ij} - b_{ij} \qquad (1 \leqq i \leqq m,\ 1 \leqq j \leqq n) \quad (7)$$

If the matrix C is the product A · B of the matrices A, B, the elements are related by the transformation

$$c_{ik} = \sum_{j=1}^{m} a_{ij} b_{jk} \qquad (1 \leqq i \leqq l,\ 1 \leq k \leq n) \quad (8)$$

During the inversion of a matrix A, it is multiplied by a sequence of matrices which convert matrix A into matrix I and convert matrix I into matrix A^{-1}. The matrix A is replaced by a matrix B which evolves into the matrix A^{-1}. When the element b_{rs} is used as the pivot for elimination the elements are replaced in accordance with the transformations

$$b_{ij} \rightarrow b_{ij} - \frac{b_{is}}{b_{rs}}\, b_{rj} \qquad (i \neq r, j \neq s) \quad (9)$$

$$b_{ij} \rightarrow - \frac{b_{is}}{b_{rs}} \qquad (i \neq r, j = s) \quad (10)$$

$$b_{ij} \rightarrow + \frac{b_{rj}}{b_{rs}} \qquad (i = r, j \neq s) \quad (11)$$

$$b_{ij} \rightarrow \frac{1}{b_{rs}} \qquad (i = r, j = s) \quad (12)$$

In the elimination process the ith row is replaced by the sum of the ith row and a fraction ϵ of the rth row where ϵ is given by the equation

$$\epsilon = - \frac{b_{is}}{b_{rs}} \qquad (13)$$

Let the partially reduced matrix be expressed by the equation

$$B = \left\| \begin{array}{ccccccccc} b_{11} & \cdots & b_{1r} & \cdots & b_{1s} & \cdots & b_{1i} & \cdots & b_{1n} \\ b_{r1} & \cdots & b_{rr} & \cdots & b_{rs} & \cdots & b_{ri} & \cdots & b_{rn} \\ b_{s1} & \cdots & b_{sr} & \cdots & b_{ss} & \cdots & b_{si} & \cdots & b_{sn} \\ b_{i1} & \cdots & b_{ir} & \cdots & b_{is} & \cdots & b_{ii} & \cdots & b_{in} \\ b_{n1} & \cdots & b_{nr} & \cdots & b_{ns} & \cdots & b_{ni} & \cdots & b_{nn} \end{array} \right\| \qquad (14)$$

When the eliminant

$$E = \left\| \begin{array}{ccccccccc} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & \epsilon & \cdots & 0 & \cdots & 1 & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 & \cdot\cdot & 0 & \cdots & 1 \end{array} \right\| \qquad (15)$$

is used as a prefactor, the ith row is replaced by the sum of the ith row and a fraction

ϵ of the rth row. When the inverse

$$E^{-1} = \left\| \begin{matrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & -\epsilon & \cdots & 0 & \cdots & 1 & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{matrix} \right\| \qquad (16)$$

is used as a postfactor, the rth column is replaced by the difference of the rth column and the fraction ϵ of the ith column. Once the elements in all rows of a column except the pivotal row are eliminated, they remain undisturbed while the elements in all rows of another column with a different pivotal row are eliminated.

During elimination the indices r, s are stored in arrays. In the index array R the indices r are stored in ascending order of the indices s, while in the index array S the indices s are stored in ascending order of the indices r.

When the indices r, s have been selected for the pivot, then the rth row and the sth column are locked out of any further consideration as a source of pivots.

If the pivots are not diagonal, the product T of the eliminants transforms the matrix A into a permutation P, and transforms the permutation P into the matrix B in accordance with the equations

$$P = T \cdot A \qquad\qquad B = T \cdot P \qquad (17)$$

The elimination of matrix T leads to the equation

$$A^{-1} = P' \cdot B \cdot P' \qquad (18)$$

where the permutation satisfies the equation

$$P^{-1} = P' \qquad (19)$$

The permutation is removed by a succession of interchanges between rows and between columns.

The values of r for two rows to be interchanged are derived from the array R. The rows are interchanged and the array is adjusted until the values of r are in ascending order. The values of s for two columns to be interchanged are derived from the array S. The columns are interchanged and the array is adjusted until the values of s are in ascending order.

Matrix arithmetic with real matrices can be adapted to matrix arithmetic with complex matrices. Let complex matrices be given by the expressions

$$A + iB \qquad\qquad C + iD \qquad (20)$$

Then complex transfer is expressed by the substitutions

$$C \rightarrow A \qquad\qquad D \rightarrow B \qquad (21)$$

Complex addition is expressed by the equation

$$(A + iB) + (C + iD) = (A + C) + (B + D)i \qquad (22)$$

Complex subtraction is expressed by the equation

$$(A + i\,B) - (C + i\,D) = (A - C) + (B - D)\,i \tag{23}$$

Complex multiplication is expressed by the equation

$$(A + i\,B) \cdot (C + i\,D) = (A \cdot C - B \cdot D) + (A \cdot D + B \cdot C)\,i \tag{24}$$

Complex inversion is expressed by the equation

$$(A + i\,B)^{-1} = (A + B \cdot A^{-1} \cdot B)^{-1} - (B + A \cdot B^{-1} \cdot A)^{-1}\,i \tag{25}$$

Thus complex matrix arithmetic can be achieved with the aid of a subroutine for real matrix arithmetic. However, complex inversion with real matrices is less efficient than direct inversion of complex matrices.

Pivot Selection

In accordance with Cramer's rule, each element occurs in all cofactors except those for the same row or column. The contribution of the element to the inverse is the product of the element and its cofactor in each determinant in which the element appears. The relative accuracy with which the element must be preserved in the inversion is relaxed when either the element or its cofactor is especially small. An optimal pivot selection would preserve the element with just whatever accuracy is required for the final inverse. The required accuracy depends not only on the nature of the matrix but also on the use of the inverse. Without prior information for each case it is necessary to adopt a general policy of pivot selection.

Various tactics have been used for pivot selection. Three options are provided by the subroutine MATRIX. In the nonpivoting option the pivots are the diagonal elements in succession. In the partial pivoting option the pivots are the largest elements in each column in succession. In the complete pivoting option the pivots are the largest elements in the remainder of the matrix.

Nonpivoting is sufficient for a matrix with diagonal dominance. Partial pivoting is inaccurate for a special matrix which has been discovered by Wilkinson[10]. Successive eliminations have the effect of doubling the size of elements in the last row and column. In the last step the subtraction of the large elements reduces them to zero or rounding error. Permutation of the columns of the matrix converts it into a Hessenberg matrix to which partial pivoting can be applied with success. Experience with many matrices has led Wilkinson to favor complete pivoting.

The relative rounding error in the elimination process is independent of any scaling of the rows or columns which does not change the sequence of pivots. If the largest element were taken as pivot, then the sequence of pivots could be upset if the selected element were diminished by a selective scaling. The matrix should be prescaled before pivot selection, or the pivot selection should be independent of scaling.

If the element b_{rs} is the pivot for the modification of the element b_{ij}, then the accuracy of the element b_{ij} is retained to within rounding error if the elements satisfy the criterion

$$0 \leq \frac{|b_{is}||b_{rj}|}{|b_{ij}||b_{rs}|} \leq 1 \tag{26}$$

On the other hand, if b_{rj} were the pivot, then the accuracy of the element b_{is} would be retained to within rounding error if the elements satisfied the criterion

$$0 \leqq \frac{|b_{ij}||b_{rs}|}{|b_{is}||b_{rj}|} \leqq 1 \tag{27}$$

Whichever quotient exceeds unity by the least amount would point to the better pivot. A symmetric criterion would be given by the difference between the quotients. Subtraction of the two quotients gives a difference which ranges from $-\infty$ to $+\infty$ and gives too much emphasis on the smallest elements where the requirements for relative accuracy can be relaxed. Less emphasis on the smallest elements is given by the criterion

$$0 \leqq \sum_{i=1}^{n} \frac{\dfrac{|b_{ij}|}{|b_{rj}|} - \dfrac{|b_{is}|}{|b_{rs}|}}{\dfrac{|b_{ij}|}{|b_{rj}|} + \dfrac{|b_{is}|}{|b_{rs}|}} \tag{28}$$

Then the quotients range in value from -1 to $+1$ and are independent of scaling. Insofar as all of the quotients for one pivot are superior to all of the quotients for the other pivot, it is necessary only to determine the sign of the criterion in order to determine which is the better pivot.

Programming

Matrix arithmetic is executed by reference to the following subroutine.

```
SUBROUTINE MTRX (MO, AA, MA, AB, MB, AC, MC)
******************************************************************************************
FORTRAN SUBROUTINE FOR MATRIX ARITHMETIC
******************************************************************************************
```

The mode of operation is given in argument MO. The matrices **A, B, C** are given or stored in the arrays AA, AB, AC. The specifications of the matrices are given in the arrays MA, MB, MC. The specification for a matrix AX consists of the array MX as given in the following table.

Address	Specification
MX(1)	Interval between columns
MX(2)	Number of rows
MX(3)	Number of columns
MX(4)	Interval between rows

The repertory of operations and call lines is given in the following table.

Operation	Call Line
$B = A$	CALL MTRX (0, AA, MA, AB, MB)
$B = A'$	CALL MTRX (1, AA, MA, AB, MB)
$C = A + B$	CALL MTRX (2, AA, MA, AB, MB, AC, MC)
$C = A - B$	CALL MTRX (3, AA, MA, AB, MB, AC, MC)
$C = A \cdot B$	CALL MTRX (4, AA, MA, AB, MB, AC, MC)
$B = A^{-1}$	CALL MTRX (5, AA, MA, DA)

During matrix inversion the matrix is replaced by its inverse and its determinant is stored in the argument DA.

PARTITIONED MATRIX ARITHMETIC

Analysis

The matrices are partitioned into submatrices by straight lines parallel to rows or columns. The numbers of rows and columns of submatrices are indicated by a sentinel record which precedes the whole matrix, while the numbers of rows and columns of elements in each submatrix are indicated by a sentinel record which precedes the submatrix. All matrices which are involved in a matrix operation must have compatible partitioning. Matrices which are to be inverted must have square diagonal submatrices. All matrices must be in the format which is compatible with binary buffer instructions.

The matrix A is an ordered rectangular array of submatrices $A_{\lambda\nu}$. When matrix A is transferred to matrix B the submatrices are related by the transformation

$$B_{\lambda\nu} \to A_{\lambda\nu} \qquad (1 \leqq \lambda \leqq m,\ 1 \leq \nu \leqq n) \quad (29)$$

When matrix A is transposed into matrix B the submatrices are related by the transformation

$$B_{\lambda\nu} \to A'_{\nu\lambda} \qquad (1 \leqq \lambda \leqq n,\ 1 \leqq \nu \leqq m) \quad (30)$$

If the matrix C is the sum A + B of the matrices A, B, the submatrices are related by the transformation

$$C_{\lambda\nu} \to A_{\lambda\nu} + B_{\lambda\nu} \qquad (1 \leqq \lambda \leqq m,\ 1 \leqq \nu \leqq n) \quad (31)$$

and if the matrix C is the difference A − B of the matrices A, B, the submatrices are related by the transformation

$$C_{\lambda\nu} \to A_{\lambda\nu} - B_{\lambda\nu} \qquad (1 \leqq \lambda \leqq m,\ 1 \leqq \nu \leqq n) \quad (32)$$

If the matrix C is the product A · B of the matrices A, B, the submatrices are related by the transformation

$$C_{\lambda\nu} \to \sum_{\mu=1}^{m} A_{\lambda\mu} \cdot B_{\mu\nu} \qquad (1 \leqq \lambda \leqq l,\ 1 \leqq \nu \leqq n) \quad (33)$$

When matrix A is inverted into matrix B by elimination with diagonal pivots the submatrices are modified in accordance with the transformations

$$B_{\lambda\nu} \to B_{\lambda\nu} - B_{\lambda\mu} \cdot B_{\mu\mu}^{-1} \cdot B_{\mu\nu} \qquad (\lambda \neq \mu, \nu \neq \mu) \quad (34)$$

$$B_{\lambda\nu} \to - B_{\lambda\mu} \cdot B_{\mu\mu}^{-1} \qquad (\lambda \neq \mu, \nu = \mu) \quad (35)$$

$$B_{\lambda\nu} \to + B_{\mu\mu}^{-1} \cdot B_{\mu\nu} \qquad (\lambda = \mu, \nu \neq \mu) \quad (36)$$

$$B_{\lambda\nu} \to B_{\mu\mu}^{-1} \qquad (\lambda = \mu, \nu = \mu) \quad (37)$$

Submatrices are read or written from tape files. Arithmetic operations on submatrices are executed in core storage.

Programming

The matrices in tape files are read into the core memory and are written from the core memory by references to subroutines. Whether the matrices are in BCD format or are in binary format is determined by whether a control index is zero or unity. Submatrices are processed sequentially or are rewound in order to eliminate wasteful backspacing. The programming must be planned so that submatrices become available when they are needed. Submatrices which are available but will not be needed until later must be stored in temporary files. Whenever submatrices are processed more than once they must be stored on a pair of tape files by references to subroutines. Which tape file is read or written is determined by whether a control index is odd or even.

Partitioned matrix arithmetic is executed by reference to the following subroutine.

```
SUBROUTINE PMTX (MO, NA, AA, MA, NB, AB, MB, NC, AC, MC, NP, NQ)
****************************************************************************************
FORTRAN SUBROUTINE FOR PARTITIONED MATRIX ARITHMETIC
****************************************************************************************
```

The mode of operation is given in argument MO. The tape file numbers for matrices A, B, C are given in the arguments NA, NB, NC. The submatrices are stored temporarily in the arrays AA, AB, AC. The specifications for submatrices are stored in the arrays MA, MB, MC. The tape file numbers for temporary storage are given in the arguments NP, NQ. Calls are made to Subroutine MTRX, Subroutine RDMTRX, Subroutine WRMTRX, Subroutine RDMXDX, and Subroutine WRMXDX.

For the transference B = A the call line to the subroutine is

```
CALL PMTX (0, NA, AA, MA, NB, AB, MB)
```

The submatrices are read from tape file NA into array AA, are transferred from array AA to array AB, and are written from array AB onto tape file NB.

For the transposition B = A' the call line to the subroutine is

```
CALL PMTX (1, NA, AA, MA, NB, AB, MB)
```

The whole matrix is read as many times as it has columns. The submatrices are read from tape file NA into array AA, the submatrices in a column are selected, and the transposed submatrices are written from array AB onto tape file NB.

For the addition C = A + B the call line to the subroutine is

```
CALL PMTX (2, NA, AA, MA, NB, AB, MB, NC, AC, MC)
```

The submatrices are read from tape files NA, NB into arrays AA, AB, and the sum is written from array AC onto tape file NC.

For the subtraction $\mathbf{C} = \mathbf{A} - \mathbf{B}$ the call line to the subroutine is

```
CALL PMTX (3, NA, AA, MA, NB, AB, MB, NC, AC, MC)
```

The submatrices are read from tape files NA, NB into arrays AA, AB, and the difference is written from array AC onto tape file NC.

For the multiplication $\mathbf{C} = \mathbf{A} \cdot \mathbf{B}$ the call line to the subroutine is

```
CALL PMTX (4, NA, AA, MA, NB, AB, MB, NC, AC, MC, NP, NQ)
```

The prefactor **A** is read once while the postfactor **B** is read as many times as there are rows in the prefactor **A**. The submatrices are read from tape files NA, NB into arrays AA, AB. The product of the submatrices is stored in array AC. If the number of columns of the prefactor is greater than one, the accumulation of the submatrices for one row are stored temporarily on tape files NP, NQ. When the accumulation of the submatrices is complete, the submatrices for one row are written from array AC onto tape file NC.

For the inversion $\mathbf{B} = \mathbf{A}^{-1}$ the call line to the subroutine is

```
CALL PMTX (5, NA, AA, MA, NB, AB, MB, NC, AC, MC, NP, NQ)
```

The matrix to be inverted is located initially on tape file NA. During transformation by Jordan elimination the successive versions of the matrix are stored alternately on the pair of tape files NA, NB. At the completion of computation the inverse matrix is stored on both tape files NA, NB. At the beginning of each cycle of elimination the submatrices in the pivotal row are written on whichever of the tape files NP, NQ is unoccupied. The pivotal submatrix is selected and its inverse is stored in array AA. The submatrices in the pivotal row are read back into array AB, are multiplied by the inverse in array AA, and the products are written from array AC onto the tape file NC. The pivotal submatrix on tape file NC is replaced by the inverse of the pivotal submatrix. Before the transformation of each row of submatrices the submatrix in the pivotal column is read initially from the first column of the original matrix, or is read subsequently from one of the tape files NP, NQ into array AA. For each elimination a submatrix is read from tape file NC into array AB, is multiplied by the submatrix in array AA, and the product is stored in array AC. The submatrix to be modified is read into array AB from one of the tape files NA, NB, and after modification it is written back out from array AB onto the other of the tape files NA, NB. In the pivotal column the submatrix to be modified is replaced by the difference between zero and the product in array AC. In the next column after the pivotal column the modified submatrix is written also on the other of the tape files NP, NQ. At the end of each cycle of elimination the submatrices are read from tape file NC and are written on the other of the tape files NA, NB.

A sentinel record and a matrix record are read from a tape file into an array by reference to the following subroutine.

```
SUBROUTINE RDMTRX (MO, NA, AA, MA)
**************************************************************************************
FORTRAN SUBROUTINE TO READ MATRIX RECORD
**************************************************************************************
```

The mode of operation is given in argument MO. The mode of operation is zero for BCD format and is unity for binary format. The tape file number is given in argument NA, the matrix is stored in the array AA, and the specification is stored in the array MA.

A sentinel record and a matrix record are written on a tape file from an array by reference to the following subroutine.

```
SUBROUTINE WRMTRX (MO, NA, AA, MA)
*****************************************************************************************
FORTRAN SUBROUTINE TO WRITE MATRIX RECORD
*****************************************************************************************
```

The mode of operation is given in argument MO. The mode of operation is zero for BCD format and is unity for binary format. The tape file number is given in argument NA, the matrix is given in the array AA, and the specification is given in the array MA.

A sentinel record and a matrix record are read from a duplex file into an array by reference to the following subroutine.

```
SUBROUTINE RDMXDX (MO, NA, NB, AA, MA)
*****************************************************************************************
FORTRAN SUBROUTINE TO READ MATRIX FROM DUPLEX FILE
*****************************************************************************************
```

The mode of operation is given in argument MO. The tape file numbers of a pair of files are given in the arguments NA, NB, the matrix is stored in the array AA, and the specification is stored in the array MA. The matrix is read from tape file NA if the mode of operation MO is odd, but the matrix is read from tape file NB if the mode of operation MO is even.

A sentinel record and a matrix record are written on a duplex file from an array by reference to the following subroutine.

```
SUBROUTINE WRMXDX (MO, NA, NB, AA, MA)
*****************************************************************************************
FORTRAN SUBROUTINE TO WRITE MATRIX ON DUPLEX FILE
*****************************************************************************************
```

The mode of operation is given in argument MO. The tape file numbers of a pair of files are given in the arguments NA, NB, the matrix is given in the array AA, and the specification is given in the array MA. The matrix is written on tape file NA if the mode of operation MO is odd, but the matrix is written on tape file NB if the mode of operation MO is even.

CHARACTERISTIC ROOTS AND VECTORS

Analysis

Regular Matrix

A covariant characteristic vector $\mathbf{a}_\mu$ of a matrix A satisfies the equation

$$(\mathrm{A} - \alpha_\mu \mathrm{I}) \cdot \mathbf{a}_\mu = 0 \qquad (38)$$

where α_μ is the μth characteristic root. A contravariant characteristic vector $\mathbf{a}^\nu$ of the matrix A satisfies the equation

$$\mathbf{a}^\nu \cdot (\mathrm{A} - \alpha_\nu \mathrm{I}) = 0 \qquad (39)$$

where α_ν is the νth characteristic root. That $\mathbf{a}^\nu$ and $\mathbf{a}_\mu$ are orthogonal if α_ν is not identical with α_μ follows from the identity

$$\mathbf{a}^\nu \cdot \mathrm{A} \cdot \mathbf{a}_\mu = \alpha_\nu \mathbf{a}^\nu \cdot \mathbf{a}_\mu = \alpha_\mu \mathbf{a}^\nu \cdot \mathbf{a}_\mu \tag{40}$$

If the characteristic roots all are distinct then the characteristic vectors are distinct and can be normalized in accordance with the equation

$$\mathbf{a}^\nu \cdot \mathbf{a}_\nu = 1 \tag{41}$$

The matrix of order n can be synthesized from idempotents in accordance with the equation

$$\mathrm{A} = \sum_{\nu=1}^{n} \alpha_\nu \mathbf{a}_\nu \mathbf{a}^\nu \tag{42}$$

If one or more of the characteristic roots are zero the matrix is singular. If the characteristic roots are equal for two or more distinct vectors, then any linear combination of the vectors is a characteristic vector. Two or more characteristic roots may be equal because their characteristic vectors are collinear, in which case the matrix is defective.

Nonzero characteristic vectors are possible only if the characteristic roots are solutions of the determinantal equation

$$p(\alpha) = |\mathrm{A} - \alpha\,\mathrm{I}| = 0 \tag{43}$$

A direct expansion of the characteristic determinant of nth order would lead to a characteristic polynomial of the nth degree. A direct expansion is feasible for matrices up to the third order, but larger matrices must be reduced to a simpler form. A prefactor S and a postfactor S^{-1} may be applied to the matrix A to convert it into a matrix B in accordance with the similarity transformation

$$\mathrm{B} = \mathrm{S} \cdot \mathrm{A} \cdot \mathrm{S}^{-1} \tag{44}$$

Then the characteristic vector $\mathbf{a}_\mu$ of matrix A is related to the characteristic vector $\mathbf{b}_\mu$ of matrix B by the equation

$$\mathbf{a}_\mu = \mathrm{S}^{-1} \cdot \mathbf{b}_\mu \tag{45}$$

and the characteristic vector $\mathbf{a}^\nu$ of matrix A is related to the characteristic vector $\mathbf{b}^\nu$ of matrix B by the equation

$$\mathbf{a}^\nu = \mathbf{b}^\nu \cdot \mathrm{S} \tag{46}$$

The characteristic roots of matrix A are undisturbed by a similarity transformation into matrix B.

Let the components of the covariant vector $\mathbf{a}_\mu$ with respect to a set of base vectors be

$$\mathbf{a}_\mu = \left\| \begin{matrix} \alpha_\mu^1 \\ \cdots \\ \cdot_\mu^n \end{matrix} \right\| \tag{47}$$

and let the components of the contravariant vector $\mathbf{a}^\nu$ with respect to the set of base vectors be

$$\mathbf{a}^\nu = \|\alpha_1^\nu, \cdots, \alpha_n^\nu\| \tag{48}$$

Orthogonality between the characteristic vectors is expressed by the equation

$$\mathbf{a}_\mu \cdot \mathbf{a}^\nu = \sum_{k=1}^{n} \alpha_\mu^k \alpha_k^\nu = \delta_\mu^\nu \tag{49}$$

where δ_μ^ν is zero when $\mu \neq \nu$ but unity when $\mu = \nu$. The component α_μ^i is the element in the ith row and the μth column of a modal matrix V, and the component α_j^ν is the element in the jth column of the νth row of the inverse matrix V^{-1}. The similarity transformation

$$\mathrm{V}^{-1} \cdot \mathrm{A} \cdot \mathrm{V} \tag{50}$$

reduces the matrix A to diagonal form.

Trianguloid Matrix

A Hessenberg or trianguloid matrix has nonzero elements on or below the diagonal next above the main diagonal, and has zero elements elsewhere above the upper diagonal.

The characteristic determinant is evaluated for any value of α by a progressive recurrence. Let Δ_m be the value of the mth principal minor in the upper left corner of the matrix. The mth principal minor is expanded into the sum of products of elements in the mth column and their cofactors. The principal minors are generated by a recurrence which starts with the equation

$$\Delta_1 = a_{11} - \alpha \tag{51}$$

and continues with the equation

$$\Delta_m = (a_{mm} - \alpha)\Delta_{m-1} - a_{m-1,m}\delta_{m-1} \tag{52}$$

where δ_{m-1} is that cofactor which is obtained by the permutation of the $(m-1)$th row and the mth row of the mth principal minor. The cofactors are generated by a recurrence which starts with the equation

$$\delta_1 = a_{m1} \tag{53}$$

and continues with the equation

$$\delta_k = a_{mk}\Delta_{k-1} - a_{k-1,k}\delta_{k-1} \tag{54}$$

until $k = m - 1$. The first derivatives of the principal minors are generated by a recurrence which starts with the equation

$$\frac{d\Delta_1}{d\alpha} = -1 \tag{55}$$

and continues with the equation

$$\frac{d\Delta_m}{d\alpha} = -\Delta_{m-1} + (a_{mm} - \alpha)\frac{d\Delta_{m-1}}{d\alpha} - a_{m-1,m}\frac{d\delta_{m-1}}{d\alpha} \tag{56}$$

The first derivatives of the cofactors are generated by a recurrence which starts with the equation

$$\frac{d\delta_1}{d\alpha} = 0 \tag{57}$$

and continues with the equation

$$\frac{d\delta_k}{d\alpha} = a_{mk} \frac{d\Delta_{k-1}}{d\alpha} - a_{k-1,k} \frac{d\delta_{k-1}}{d\alpha} \tag{58}$$

until $k = m - 1$. The second derivatives of the principal minors are generated by a recurrence which starts with the equation

$$\frac{d^2\Delta_1}{d\alpha^2} = 0 \tag{59}$$

and continues with the equation

$$\frac{d^2\Delta_m}{d\alpha^2} = -2 \frac{d\Delta_{m-1}}{d\alpha} + (a_{mm} - \alpha) \frac{d^2\Delta_{m-1}}{d\alpha^2} - a_{m-1,m} \frac{d^2\delta_{m-1}}{d\alpha^2} \tag{60}$$

The second derivatives of the cofactors are generated by a recurrence which starts with the equation

$$\frac{d^2\delta_1}{d\alpha^2} = 0 \tag{61}$$

and continues with the equation

$$\frac{d^2\delta_k}{d\alpha^2} = a_{mk} \frac{d^2\Delta_{k-1}}{d\alpha^2} - a_{k-1,k} \frac{d^2\delta_{k-1}}{d\alpha^2} \tag{62}$$

until $k = m - 1$. The nth order principal minor Δ_n is the characteristic determinant $|\mathrm{A} - \alpha\mathrm{I}|$.

The determinant and its derivatives are utilized by a general routine for the determination of the real and complex roots of the determinantal polynomial $p(\alpha)$.

The product of the characteristic matrix and a characteristic vector is zero for any characteristic root. Sequential evaluation of the elements of the characteristic vector v_μ shows that the ith element is given by the equation

$$v_\mu^i = -\frac{1}{a_{i-1,i}} \left\{ \sum_{k=1}^{i-1} a_{i-1,k} v_\mu^k - \alpha_\mu v_\mu^{i-1} \right\} \tag{63}$$

The recurrence for v_μ^i starts with $v_\mu^1 = 1$ and defines uniquely the sequence of elements for v_μ as long as the pivotal element $a_{i-1,i}$ is nonzero. If the pivotal element vanishes then all elements with index less than i are set equal to zero unless the summation also is zero. Sequential evaluation of the elements of the characteristic vector v^ν shows that the jth element is given by the equation

$$v_j^\nu = -\frac{1}{a_{j,j+1}} \left\{ \sum_{k=j+1}^{n} a_{k,j+1} v_k^\nu - \alpha_\nu v_{j+1}^\nu \right\} \tag{64}$$

The recurrence for v_j^ν starts with $v_n^\nu = 1$ and defines uniquely the sequence of elements for v^ν as long as the pivotal element $a_{j,j+1}$ is nonzero. If the pivotal element vanishes then all elements with index more than j are set equal to zero unless the summation also is zero.

Two vectors are computed by back substitution. The first element of one vector is unity and the first element of the other vector is zero. Each element of the vectors is the quotient of a summation and a pivot. Whenever the pivot is zero, the first vector is replaced by that linear combination of the two vectors which reduces the summation to zero, and the second vector is set equal to zero. The next element then is set equal to zero in the first vector and equal to unity in the second vector. Finally the first vector is replaced by that linear combination of the two vectors which satisfies the last row of the matrix.

Trianguloidization

In the trianguloidization of a matrix by a similarity transformation the matrix is reduced one column at a time. For each successive diagonal element, the next element above in the same column is used as a pivot to eliminate all remaining nonzero elements above. It is required that once elements are reduced to zero in any cycle of transformation they shall remain zero in all subsequent cycles.

Let the partially transformed matrix be expressed by the equation

$$B = \begin{Vmatrix} b_{11} & \cdots & b_{1j} & \cdots & b_{1s} & \cdots & 0 \\ b_{i1} & \cdots & b_{ij} & \cdots & b_{is} & \cdots & 0 \\ b_{s1} & \cdots & b_{sj} & \cdots & b_{ss} & \cdots & 0 \\ b_{n1} & \cdots & b_{nj} & \cdots & b_{ns} & \cdots & b_{nn} \end{Vmatrix} \tag{65}$$

where $n - s$ columns have been trianguloidized. The elements in the ith row are replaced in accordance with the transformation

$$b_{ij} \to b_{ij} - \frac{b_{is}}{b_{s-1,s}} b_{s-1,j} \qquad (j > s + 1) \quad (66)$$

After elimination by subtraction of a fraction of the $(s - 1)$th row from the ith row, the same fraction of the ith column is added to the $(s - 1)$th column. The elements in the $(s - 1)$th column are replaced in accordance with the transformation

$$b_{k,s-1} \to b_{k,s-1} + \frac{b_{is}}{b_{s-1,s}} b_{ki} \tag{67}$$

The accuracy of the elements is retained to within rounding error if the elements satisfy the criteria

$$0 \leqq \frac{|b_{is}||b_{s-1,j}|}{|b_{ij}||b_{s-1,s}|} \leqq 1 \tag{68}$$

and

$$0 \leqq \frac{|b_{is}||b_{ki}|}{|b_{s-1,s}||b_{k,s-1}|} \leqq 1 \tag{69}$$

The pivot $b_{s-1,s}$ is superior to the pivot b_{is} if the pivots satisfy the criterion

$$0 \leqq \sum_{j=1}^{n} \frac{\dfrac{|b_{ij}|}{|b_{is}|} - \dfrac{|b_{s-1,j}|}{|b_{s-1,s}|}}{\dfrac{|b_{ij}|}{|b_{is}|} + \dfrac{|b_{s-1,j}|}{|b_{s-1,s}|}} + \sum_{k=1}^{n} \frac{\dfrac{|b_{k,s-1}|}{|b_{is}|} - \dfrac{|b_{ki}|}{|b_{s-1,s}|}}{\dfrac{|b_{k,s-1}|}{|b_{is}|} + \dfrac{|b_{ki}|}{|b_{s-1,s}|}} \tag{70}$$

The similarity transformation requires that for any scaling of the ith row there must be inverse scaling of the ith column. Within this limitation on scaling the pivot selection is independent of scaling.

Programming

SUBROUTINE RVMTX (AA, NA, RA, VA, SA)

****

FORTRAN SUBROUTINE FOR ROOTS AND VECTORS OF A MATRIX

****

The address of the matrix A is the $n \times n$ array AA. The order n of the matrix is given in the argument NA. The matrix is trianguloidized by permutations and eliminations. The real and the imaginary parts of the characteristic roots are stored in alternate addresses of the $2n$-array RA. The covariant vectors of the matrix are stored columnwise with the real and the imaginary parts of each vector in alternate columns of the $n \times 2n$ array VA. The similarity transformation for the trianguloidization is stored in the $n \times n$ array SA. References are made to Subroutine MXRT for the determination of roots.

SUBROUTINE CHDMT (MO, AA, DA, DD, SD, AM, NM, PM, DM, SM)

****

FORTRAN SUBROUTINE FOR CHARACTERISTICS OF TRIANGULOID MATRIX

****

The mode of operation is given in the argument MO. The real and the imaginary parts of the argument α are given in the 2-array AA. If MO is 1, 2, 3 the computations are carried as far as the determinant, its first derivative, or its second derivative. The determinant, its first derivative, and its second derivative are stored in the 2-arrays DA, DD, SD. The address of the matrix A is the $n \times n$ array AM. The order of the matrix is given in the argument NM. The principal minor determinants and their first and second derivatives are stored in the $2n$-arrays PM, DM, SM.

SUBROUTINE MXRT (CD, CE, AZ, NA, AA, AM, PM, DM, SM)

****

FORTRAN SUBROUTINE FOR CHARACTERISTIC ROOTS OF MATRIX

****

The step length δ for hunting is given in argument CD, and the tolerance ϵ for homing is given in argument CE. The initial position z is given in the 2-array AZ. The number of roots n is given in argument NA. The address of the matrix A is the $n \times n$ array AM. The addresses of principal minors and their first and second derivatives are the $2n$-arrays PM, DM, SM. The region of a root is located by a hunting procedure with steps of fixed length, then the root is determined by a homing procedure with Newton-Raphson iteration. References are made to Subroutine CHDMT to obtain values of the characteristic determinant and its first and second derivatives. The real and imaginary parts of the roots are stored in alternate addresses of the $2n$-array AA.

SUBROUTINE OMTXV (NA, VL, VR)

****

FORTRAN SUBROUTINE FOR ORTHONORMAL VECTORS OF MATRIX

****

The order n of the matrix is given in argument NA. The left and the right vectors are given in the $n \times 2n$ arrays VL and VR. All vectors after a pivotal vector in one set are orthogonalized progressively with respect to the pivotal vector in the other set, until each vector of one set is orthogonal to the other vectors of the other set. Finally, each vector of one set is normalized with respect to the same vector of the other set in such a way that the largest elements in either vector have the same magnitude and the same or opposite direction.

DISCUSSION

The fact that FORTRAN requires matrices to be stored in transposed form is connected with the choice of indices for two-index arrays. In FORTRAN the first index is the most rapidly variable, whereas it would be more natural for the last index to be the most rapidly variable. The designers of FORTRAN made the wrong choice.

The conjugate-gradient method was presented originally by Hestenes and Stiefel[8] for symmetric positive definite matrices. A generalization for asymmetric matrices is derived in Appendix A. Unfortunately, the conjugate-gradient method is not suitable for a general purpose subroutine because rounding errors cause rapid deterioration of the recurrence relations.

Rotations have been used by Givens[6] and reflections have been used by Householder[7] to reduce a general matrix to triangular form. That Gauss elimination is more efficient for triangularization has been reported by Wilkinson.

The method of minimized iterations was presented originally by Lanczos[9] for asymmetric matrices. The algorithm is rederived in Appendix B. Unfortunately, the method of minimized iterations is not suitable for a general purpose subroutine because rounding errors in nearly zero divisors upset the recurrence relations.

Rotations have been used by Givens[12] and reflections have been used by Householder[12] to reduce a symmetric matrix to tridiagonal form. A generalization of the Householder method to nonsymmetric matrices is derived in Appendix B. The method fails if the inner product of a row and a column becomes zero.

The figure of merit of a matrix subroutine is the number of matrices in a set of test matrices which the subroutine is able to process successfully. There was a set of test matrices on NORC, and there is a set of test matrices in a book by Gregory and Karney[14]. Matrices from both sets have been collected into a single set by N. M. Wolcott. The matrices which are defined for arbitrary order are cataloged in Appendix C. To these have been added special matrices to make a set of fifty test matrices of the sixth order.

Extensively used in this laboratory is a subroutine MINVR, which uses Gauss-Jordan elimination with complete pivot selection. The matrix must be compact but is inverted in place. The CDC 6600 subroutine MATRIX inverts a matrix in place by Gauss-Jordan elimination with a choice of no pivoting, partial pivoting, or complete pivoting. The matrix must be compact but may be a submatrix of a larger matrix. The IBM 360 subroutine MINV inverts a matrix in place by Gauss-Jordan elimination with complete pivoting. The matrix must be compact. The IMSL subroutine LINV1F uses the Crout algorithm to solve a system of simultaneous equations for which the known elements are the columns of the identity matrix. The space for two matrices is required, one to hold the original matrix and one to hold the identity matrix. The matrices must be compact, but may be submatrices of larger matrices. A subroutine CROUT has been prepered by A. H. Morris, Jr. It uses the Crout algorithm with partial pivoting to invert a matrix in place. The matrix must be compact, but may be a submatrix of a larger matrix. The accumulation of inner products is in double precision.

It is known that for matrix inversion the Crout algorithm requires the same number of operations as the Gauss–Jordan algorithm. The accumulation of each sum of inner products occurs in a single operation in the Crout method and more accuracy is possible in a computer with a double-precision accumulator. When the two methods are programmed in FORTRAN, their relative efficiencies depend upon the extent to which the compiler takes advantage of parallel processing in the central processing unit.

In a series of test runs the six subroutines were applied to the inversion of the fifty test matrices. The subroutine MATRIX ran the fastest because it is written in COMPASS. The subroutines CROUT and MTRX were nearly equally fast. The accuracy of inversion was determined from the product of each matrix with its inverse. The extent to which the product deviates from the identity matrix is an indication of the accuracy of inversion. All of the subroutines gave comparable accuracy. CROUT was a fraction of a digit better than MTRX for most matrices, but MTRX was better than CROUT for a few matrices.

The CDC subroutine MATRIX uses the Householder method, and the IBM subroutine EIGEN uses the Jacobi method to find the roots and vectors of a real symmetric matrix. However, symmetric matrices are not within the scope of the present investigation. The IMSL subroutine EIGRF and the EISPACK subroutines use the QR algorithm. A driver EIGV for the EISPACK subroutines has been prepared by A. H. Morris, Jr.

In a series of test runs the three sets of subroutines were applied to the computation of the roots and vectors of the fifty test matrices. The subroutines EIGRF and EIGV ran more than three times as fast as RVMTX. The accuracy of computation was determined from the difference between the products of the matrix and its vectors and the products of the vectors and their roots. With normalization of the vectors to unit magnitude, the accuracy of RVMTX was one digit better than the accuracy of EIGRF and EIGV for most of the matrices. Both routines gave repeated vectors for the defective matrices. The roots and vectors are sorted in ascending order by RVMTX, but the roots and vectors are not arranged in any order by EIGV.

CONCLUSION

Pivot selection with scale invariance gives as much accuracy as full pivot selection in single precision subroutines. The determination of roots and vectors by Newton–Raphson iteration gives one digit better accuracy than the determination by blockwise triangularization.

BIBLIOGRAPHY

1. *Control Data* 6000 *Series Computer Systems. Matrix Algebra Subroutines Reference Manual.*
   Control Data Corporation Publication No. 60135200, (Control Data Corporation, Minneapolis, Minnesota, 1966)

2. *System/*360 *Scientific Subroutine Package.*
   Programmer's Manual Number 360A–CM–03X, (International Business Machines Corporation, New York, 1970)

3. *IMSL Library* 3 *Reference Manual.*
   (International Mathematical and Statistical Libraries, Inc., Houston, Texas, 1977)

4. *Lecture Notes in Computer Science, Volume* 6.
   B. T. Smith, J. M. Boyle, J. J. Dongarra, B. S. Garbow, Y. Ikebe, V. C. Klema, C. B. Moler, Matrix Eigensystem Routines–EISPACK Guide. (Springer–Verlag, New York, 1974)

5. *Basic Theorems in Matrix Theory.*
   M. Marcus, National Bureau of Standards Applied Mathematics Series No. 57 (January 1960)

6. *Computation of Plane Unitary Rotations Transforming a General Matrix to Triangular Form.*
   W. Givens, Journal of the Society for Industrial and Applied Mathematics, **6**, 26 (1958)

7. *Unitary Triangularization of a Nonsymmetric Matrix.*
   A. S. Householder, Journal of the Association for Computing Machinery, **5**, 339 (1958)

8. *Methods of Conjugate Gradients for Solving Linear Systems.*
   M. R. Hestenes, and E. Stiefel, Journal of Research of the National Bureau of Standards, **49**, 408 (1952)

9. *An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators.*
   C. Lanczos, Journal of Research of the National Bureau of Standards, **45**, 255 (1950)

10. *Error Analysis of Direct Methods of Matrix Inversion.*
    J. H. Wilkinson, Journal of the Association for Computing Machinery, **8**, 281 (1961)

11. *Rounding Errors in Algebraic Processes.*
    J. H. Wilkinson, (Prentice–Hall, Inc., Englewood Cliffs, N. J. 1963)

12. *The Algebraic Eigenvalue Problem.*
    J. H. Wilkinson, (Oxford University Press, 1965)

13. *Handbook for Automatic Computation, Volume II. Linear Algebra.*
    J. H. Wilkinson, and C. Reinsch, (Springer–Verlag, New York, 1971)

14. *A Collection of Matrices for Testing Computational Algorithms.*
    R. T. Gregory, and D. L. Karney, (John Wiley and Sons, New York, 1969)

APPENDIX A

MATRIX INVERSION

CONJUGATE GRADIENT

Let a circle be inscribed in a square and let the square and the circle be deformed by a linear transformation into a parallelogram and an ellipse. The points of tangency between parallelogram and ellipse are opposite ends of conjugate diameters. One conjugate diameter is orthogonal to the normal to the ellipse at the end of the other conjugate diameter. If the end of the diameter and the normal to the ellipse are given, then the center of the ellipse is in the direction of the diameter. The ellipse is the contour of constant value for a quadratic function of the coordinates. The quadratic function is zero at the center of the ellipse.

Let x be an unknown vector and let y be a known vector. Let r be the residual which is defined by the equation

$$\mathbf{r} = \mathbf{y} - \mathbf{A} \cdot \mathbf{x} \tag{1}$$

The vector x is a solution of the equation

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \tag{2}$$

at the point where the quadratic function $\mathbf{r} \cdot \mathbf{r}$ is zero. The normal to the contour of constant $\mathbf{r} \cdot \mathbf{r}$ is in the direction of the gradient

$$\nabla(\mathbf{r} \cdot \mathbf{r}) = -2\mathbf{A}' \cdot \mathbf{r} \tag{3}$$

which defines the direction conjugate to the vector r.

In the method of Hestenes and Stiefel, the solution is approached by a sequence of vectors $\mathbf{x}_1, \cdots, \mathbf{x}_n$ which minimize the function $\mathbf{r} \cdot \mathbf{r}$. Associated with the vectors is a sequence of residuals $\mathbf{r}_1, \cdots, \mathbf{r}_n$ and a sequence of gradients $\mathbf{A}' \cdot \mathbf{r}_1, \cdots, \mathbf{A}' \cdot \mathbf{r}_n$. Let the vector $\mathbf{p}_k$ be a linear combination of the first k of the vectors $\mathbf{A}' \cdot \mathbf{r}_1, \cdots, \mathbf{A}' \cdot \mathbf{r}_n$. Let the $(k+1)$th vector $\mathbf{x}_{k+1}$ be expressed in terms of the kth vector $\mathbf{x}_k$ by the equation

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \tag{4}$$

where α_k is a scalar. The residuals are related in accordance with the equation

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A} \cdot \mathbf{p}_k \tag{5}$$

The square of the residual is given by the equation

$$\mathbf{r}_{k+1} \cdot \mathbf{r}_{k+1} = \mathbf{r}_k \cdot \mathbf{r}_k - 2\alpha_k \mathbf{p}_k \cdot \mathbf{A}' \cdot \mathbf{r}_k + \alpha_k^2 \mathbf{p}_k \cdot \mathbf{A}' \cdot \mathbf{A} \cdot \mathbf{p}_k \tag{6}$$

The square of the residual is a minimum where the derivative with respect to α_k is zero, or where α_k is given by the equation

$$\alpha_k = \frac{\mathbf{p}_k \cdot \mathbf{A}' \cdot \mathbf{r}_k}{\mathbf{p}_k \cdot \mathbf{A}' \cdot \mathbf{A} \cdot \mathbf{p}_k} \tag{7}$$

Substitution of this value of α_k in the expression for $\mathbf{r}_{k+1}$ and multiplication by $\mathbf{p}_k \cdot \mathbf{A}'$ shows that

$$\mathbf{p}_k \cdot \mathbf{A}' \cdot \mathbf{r}_{k+1} = 0 \tag{8}$$

Let the vector $\mathbf{p}_k$ be expressed by the equation

$$\mathbf{p}_k = \mathrm{A}' \cdot \mathbf{r}_k + \sum_{m=1}^{k-1} \beta_m \mathbf{p}_m \tag{9}$$

where β_m is a scalar. Let the vectors be orthogonal as expressed by the equation

$$\mathbf{p}_m \cdot \mathrm{A}' \cdot \mathrm{A} \cdot \mathbf{p}_k = 0 \qquad (m \neq k) \ (10)$$

Then β_m is given by the equation

$$\beta_m = -\frac{\mathbf{p}_m \cdot \mathrm{A}' \cdot \mathrm{A} \cdot \mathrm{A}' \cdot \mathbf{r}_k}{\mathbf{p}_m \cdot \mathrm{A}' \cdot \mathrm{A} \cdot \mathbf{p}_m} \tag{11}$$

and the vectors are constructed by a multi-term recurrence.

For any $m < k$ the residual $\mathbf{r}_{k+1}$ can be expanded in the polynomial

$$\mathbf{r}_{k+1} = \mathbf{r}_{m+1} - \alpha_{m+1}\mathrm{A} \cdot \mathbf{p}_{m+1} - \cdots - \alpha_k \mathrm{A} \cdot \mathbf{p}_k \tag{12}$$

Multiplication by $\mathbf{p}_m \cdot \mathrm{A}'$ leads to the equation

$$\mathbf{p}_m \cdot \mathrm{A}' \cdot \mathbf{r}_{k+1} = 0 \tag{13}$$

Thus the square of the residual is a minimum with respect to any $\delta\mathbf{x}$ given by the equation

$$\delta\mathbf{x} = \sum_{m=1}^{k} \mathbf{p}_m \, \delta\alpha_m \tag{14}$$

The square of the residual is reduced to zero after n steps.

For any $m < k$ the vectors are related by the equation

$$\mathbf{r}_k \cdot \mathrm{A} \cdot \mathbf{p}_m = 0 \qquad (m < k) \ (15)$$

or for $m = k$ by the equation

$$\mathbf{r}_k \cdot \mathrm{A} \cdot \mathbf{p}_k = \mathbf{r}_k \cdot \mathrm{A} \cdot \mathrm{A}' \cdot \mathbf{r}_k \qquad (m = k) \ (16)$$

and for $m \neq k$ by the equation

$$\mathbf{r}_k \cdot \mathrm{A} \cdot \mathrm{A}' \cdot \mathbf{r}_m = \mathbf{r}_m \cdot \mathrm{A} \cdot \mathrm{A}' \cdot \mathbf{r}_k = 0 \qquad (m \neq k) \ (17)$$

Substitution for $\alpha_m \mathrm{A} \cdot \mathbf{p}_m$ its expression in terms of residuals leads to the equation

$$\beta_m = -\frac{(\mathbf{r}_{m+1} - \mathbf{r}_m) \cdot \mathrm{A} \cdot \mathrm{A}' \cdot \mathbf{r}_k}{(\mathbf{r}_{m+1} - \mathbf{r}_m) \cdot \mathrm{A} \cdot \mathbf{p}_m} \tag{18}$$

from which it is apparent that $\beta_m = 0$ for all $m < k - 1$. Thus the vectors $\mathbf{p}_k$ are synthesized by a two-term recurrence equation

$$\mathbf{p}_k = \mathrm{A}' \cdot \mathbf{r}_k + \beta_{k-1}\mathbf{p}_{k-1} \tag{19}$$

for which β_{k-1} is given by the equation

$$\beta_{k-1} = \frac{\mathbf{r}_k \cdot \mathrm{A} \cdot \mathrm{A}' \cdot \mathbf{r}_k}{\mathbf{r}_{k-1} \cdot \mathrm{A} \cdot \mathrm{A}' \cdot \mathbf{r}_{k-1}} \tag{20}$$

The effect of rounding error on the mutual orthogonality of the vectors tends to grow rapidly with the two-term recurrence, but can be checked with the multi-term recurrence.

Inasmuch as the vectors $\mathrm{A} \cdot \mathbf{p}_k$ are mutually orthogonal, they can be used to express the identity matrix by the equation

$$\mathbf{I} = \sum_{k=1}^{n} \frac{\mathbf{p}_k \cdot \mathbf{A}'\mathbf{A} \cdot \mathbf{p}_k}{\mathbf{p}_k \cdot \mathbf{A}' \cdot \mathbf{A} \cdot \mathbf{p}_k} \tag{21}$$

whence the inverse matrix is given by the equation

$$\mathbf{A}^{-1} \approx \sum_{k=1}^{n} \frac{\mathbf{p}_k\mathbf{A} \cdot \mathbf{p}_k}{\mathbf{p}_k \cdot \mathbf{A}' \cdot \mathbf{A} \cdot \mathbf{p}_k} \tag{22}$$

The accuracy of the inverse matrix is limited by the extent to which the initial residual vector has substantial components in the directions of all of the characteristic vectors of the matrix.

APPENDIX B

CHARACTERISTIC

ROOTS AND VECTORS

COMPANION MATRIX

A companion matrix has ones on the diagonal next above the main diagonal, nonzero elements along the bottom row, and zero elements elsewhere. The characteristic equation of the matrix A is

$$(-1)^n|\mathrm{A} - \alpha\, \mathrm{I}| = p(\alpha) = \sum_{m=0}^{n} c_m \alpha^m \tag{1}$$

where the nth order coefficient of the characteristic polynomial $p(\alpha)$ is unity. The characteristic equation of a companion matrix is

$$|\mathrm{A} - \alpha\, \mathrm{I}| = \begin{vmatrix} -\alpha & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -\alpha & 1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & -\alpha & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & -\alpha & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -\alpha & 1 \\ -c_0 & -c_1 & -c_2 & \cdots & -c_{n-3} & -c_{n-2} & -c_{n-1}-\alpha \end{vmatrix} \tag{2}$$

That the characteristic determinant is equal to $(-1)^n p(\alpha)$ may be verified if each column on the right is multiplied by α and is added to the next column on the left until diagonal elements above the bottom row are eliminated, and columns are interchanged to bring the first column into the position of the last column. Then the matrix of the determinant is triangular and the determinant is equal to the product of its diagonal elements, which are all ones except the last element.

The product of the characteristic matrix and a characteristic vector is zero for any characteristic root. Sequential evaluation of the elements of the characteristic vector v_μ shows that the ith element is given by the equation

$$\mathrm{v}_\mu = \begin{Vmatrix} 1 \\ \cdots \\ \alpha_\mu^{i-1} \\ \cdots \\ \alpha_\mu^{n-1} \end{Vmatrix} \tag{3}$$

Sequential evaluation of the elements of the characteristic vector v^ν shows that the jth element is given by the equation

$$\mathrm{v}^\nu = \|p_{n-1}(\alpha_\nu), \cdots, p_{n-j}(\alpha_\nu), \cdots, 1\| \tag{4}$$

where $p_k(\alpha_\nu)$ is just the kth order polynomial which is constructed during the nested method of polynomial evaluation.

If all characteristic roots are distinct, then the vectors $\mathbf{v}_\mu$ are proportional to the columns of a modal matrix V, and the vectors $\mathbf{v}^\nu$ are proportional to the rows of the inverse matrix V^{-1}.

REDUCTION TO COMPANION

In the Danilevsky reduction a matrix is transformed into its companion matrix by a sequence of eliminations. The first element above each diagonal element is used as a pivot to eliminate all other elements in the same row. Let the partially transformed matrix be expressed by the equation

$$\mathrm{B} = \begin{Vmatrix} 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ b_{s1} & \cdots & b_{ss} & b_{s,s+1} & \cdots & b_{sn} \\ b_{s+1,1} & \cdots & b_{s+1,s} & b_{s+1,s+1} & \cdots & b_{s+1,n} \\ b_{n1} & \cdots & b_{ns} & b_{n,s+1} & \cdots & b_{nn} \end{Vmatrix} \tag{5}$$

where $s-1$ rows of the matrix have been reduced to upper diagonal form. The elements in the jth column are replaced in accordance with the transformation

$$b_{ij} \rightarrow b_{ij} - \frac{b_{sj}}{b_{s,s+1}} b_{i,s+1} \tag{6}$$

After elimination by subtraction of a fraction of the $(s+1)$th column from the jth column, the same fraction of the jth row is added to the $(s+1)$th row. The elements in the $(s+1)$th row are replaced in accordance with the transformation

$$b_{s+1,k} \rightarrow b_{s+1,k} + \frac{b_{sj}}{b_{s,s+1}} b_{jk} \tag{7}$$

After elimination the $(s+1)$th column is divided by $b_{s,s+1}$ and the $(s+1)$th row is multiplied by $b_{s,s+1}$.

The method of reduction fails for a diagonal matrix.

Even after the characteristic polynomial has been derived by reduction of the matrix, a polynomial of high degree cannot be evaluated for large arguments, and the largest roots are indeterminate.

TRIDIAGONALIZATION

In the tridiagonalization of a matrix by a similarity transformation the matrix is tridiagonalized progressively one row and one column at a time. At each successive diagonal element, the next right element in the same row, and the next lower element in the same column are used as pivots to eliminate the remaining nonzero elements in the row and the column. It is required that once elements are reduced to zero in any cycle of transformation they shall remain zero in all subsequent cycles.

Let the partially transformed matrix be expressed by the equation

$$B = \left\| \begin{array}{ccccc} b_{11} & 0 & 0 & 0 & 0 \\ 0 & b_{ss} & b_{s,s+1} & b_{sj} & b_{sn} \\ 0 & b_{s+1,s} & b_{s+1,s+1} & \cdots & \cdots \\ 0 & b_{is} & \cdots & \cdots & \cdots \\ 0 & b_{ns} & \cdots & \cdots & b_{nn} \end{array} \right\| \tag{8}$$

where $s - 1$ rows and columns have been tridiagonalized.

The Householder tridiagonalization is based on the application of a similarity transformation in which the prefactor and the postfactor both are of the same form,

$$I - 2\frac{uv}{u \cdot v} \tag{9}$$

The prefactor and the postfactor satisfy the identity

$$\left(I - 2\frac{uv}{u \cdot v}\right) \cdot \left(I - 2\frac{uv}{u \cdot v}\right) = I \tag{10}$$

The application of this similarity transformation to matrix B is expressed by the equation

$$\left(I - 2\frac{uv}{u \cdot v}\right) \cdot B \cdot \left(I - 2\frac{uv}{u \cdot v}\right) = B - 2\frac{u(v \cdot B)}{u \cdot v} - 2\frac{(B \cdot u)v}{u \cdot v} + 4\frac{(v \cdot B \cdot u)uv}{(u \cdot v)^2} \tag{11}$$

If all elements of index less than s - 1 are zero in both u and v, then uv contains no elements in the sth column or sth row, but u(v · B) contains elements in the sth column, while (B · u)v contains elements in the sth row. For elements below and to the right of the pivotal elements to become and remain zero, it is necessary for the elements of u and v beyond the pivotal element to be proportional to the column or the row which is to be eliminated.

Let the vector u be the column vector

$$u = \left\| \begin{array}{c} 0 \\ \cdots \\ 0 \\ q_{s+1,s} \\ b_{s+2,s} \\ \\ b_{ns} \end{array} \right\| \tag{12}$$

and let the vector v be the row vector

$$v \quad 0, \quad , 0, q_{s,s+1}, b_{s,s+2}, \quad , b_{sn} \tag{13}$$

The scalar product of u and v is given by the equation

$$\mathbf{u}\cdot\mathbf{v} = q_{s,s+1}q_{s+1,s} + \sum_{k=s+2}^{n} b_{sk}b_{ks} \tag{14}$$

The *i*th element of the *s*th column becomes

$$b_{is} - 2\,\frac{q_{s,s+1}b_{s+1,s} + \sum\limits_{k=s+2}^{n} b_{sk}b_{ks}}{q_{s,s+1}q_{s+1,s} + \sum\limits_{k=s+2}^{n} b_{sk}b_{ks}}\, b_{is} \qquad (i \neq s+1) \tag{15}$$

Thus the elements in the *s*th column are eliminated if the parameters $q_{s,s+1}$ and $q_{s+1,s}$ satisfy the equation

$$q_{s,s+1}q_{s+1,s} - 2q_{s,s+1}b_{s+1,s} - \sum_{k=s+2}^{n} b_{sk}b_{ks} = 0 \tag{16}$$

The *j*th element of the *s*th row becomes

$$b_{sj} - 2\,\frac{b_{s,s+1}q_{s+1,s} + \sum\limits_{k=s+2}^{n} b_{sk}b_{ks}}{q_{s,s+1}q_{s+1,s} + \sum\limits_{k=s+2}^{n} b_{sk}b_{ks}}\, b_{sj} \qquad (j \neq s+1) \tag{17}$$

Thus the elements in the *s*th row are eliminated if the parameters $q_{s,s+1}$ and $q_{s+1,s}$ satisfy the equation

$$q_{s,s+1}q_{s+1,s} - 2b_{s,s+1}q_{s+1,s} - \sum_{k=s+2}^{n} b_{sk}b_{ks} = 0 \tag{18}$$

Comparison of the equations for the parameters shows that they are reduced to a single equation by the substitutions

$$q_{s+1,s} = \eta b_{s+1,s} \qquad\qquad q_{s,s+1} = \eta b_{s,s+1} \tag{19}$$

where the parameter η is a solution of the equation

$$\eta^2 - 2\eta - \frac{\sum\limits_{k=s+2}^{n} b_{sk}b_{ks}}{b_{s,s+1}b_{s+1,s}} = 0 \tag{20}$$

Thus η is given by the equation

$$\eta = 1 \pm \left[\frac{\sum\limits_{k=s+1}^{n} b_{sk}b_{ks}}{b_{s,s+1}b_{s+1,s}}\right]^{\frac{1}{2}} \tag{21}$$

and u · v is given by the equation

$$\mathbf{u}\cdot\mathbf{v} = \pm\, 2(b_{s,s+1}b_{s+1,s})^{\frac{1}{2}}\left(\sum_{k=s+1}^{n} b_{sk}b_{ks}\right)^{\frac{1}{2}} + 2\sum_{k=s+1}^{n} b_{sk}b_{ks} \tag{22}$$

If the product of the pivotal elements happens to be zero, then rows and columns can be permuted to bring into pivotal position a pair of elements with a finite product. The value of η is real only if the product $b_{s,s+1}b_{s+1,s}$ has the same sign as the sum

$$\sum_{k=s+1}^{n} b_{sk}b_{ks} \tag{23}$$

There always must be at least one pair of elements for which the product of the pair has the same sign as the sum of products. If there is more than one the pair with the product of largest magnitude is selected for pivotal elements. The sign of the radical is selected to give $\mathbf{u} \cdot \mathbf{v}$ the larger magnitude.

The method fails if the sum of the products is zero as expressed by the equation

$$\sum_{k=s+1}^{n} b_{sk}b_{ks} = 0 \tag{24}$$

Then no permutations or eliminations can prevent $\mathbf{u} \cdot \mathbf{v}$ from having a value of zero. The method fails for a companion matrix, which may have distinct roots and vectors. The companion matrix contains information only about roots and not about the vectors of any matrix from which the companion matrix may have been derived.

MINIMIZED ITERATION

In the method of minimized iteration, the roots and vectors are determined from a sequence of biorthogonal vectors in which each successive vector is derived from previous vectors through multiplication by the matrix A.

Let $\mathbf{j}_0, \cdots, \mathbf{j}_{n-1}$ and $\mathbf{j}^0, \cdots, \mathbf{j}^{n-1}$ be biorthogonal vectors such that

$$\mathbf{j}_\mu = \frac{1}{\mathbf{j}^\mu \cdot \mathrm{A} \cdot \mathbf{j}_{\mu-1}} \left(\mathrm{A} \cdot \mathbf{j}_{\mu-1} - \sum_{\nu=0}^{\mu-1} \frac{\mathbf{j}_\nu \mathbf{j}^\nu \cdot \mathrm{A} \cdot \mathbf{j}_{\mu-1}}{\mathbf{j}_\nu \cdot \mathbf{j}^\nu} \right) \tag{25}$$

and

$$\mathbf{j}^\mu = \frac{1}{\mathbf{j}^{\mu-1} \cdot \mathrm{A} \cdot \mathbf{j}_\mu} \left(\mathbf{j}^{\mu-1} \cdot \mathrm{A} - \sum_{\nu=0}^{\mu-1} \frac{\mathbf{j}^{\mu-1} \cdot \mathrm{A} \cdot \mathbf{j}_\nu \mathbf{j}^\nu}{\mathbf{j}_\nu \cdot \mathbf{j}^\nu} \right) \tag{26}$$

Multiplication by $\mathbf{j}^\lambda$ is given by the equation

$$\mathbf{j}^\lambda \cdot \mathbf{j}_\mu = \frac{1}{\mathbf{j}^\mu \cdot \mathrm{A} \cdot \mathbf{j}_{\mu-1}} \left(\mathbf{j}^\lambda \cdot \mathrm{A} \cdot \mathbf{j}_{\mu-1} - \sum_{\nu=0}^{\mu-1} \frac{\mathbf{j}^\lambda \cdot \mathbf{j}_\nu \mathbf{j}^\nu \cdot \mathrm{A} \cdot \mathbf{j}_{\mu-1}}{\mathbf{j}_\nu \cdot \mathbf{j}^\nu} \right) \tag{27}$$

and multiplication by $\mathbf{j}_\lambda$ is given by the equation

$$\mathbf{j}^\mu \cdot \mathbf{j}_\lambda = \frac{1}{\mathbf{j}^\mu \cdot \mathrm{A} \cdot \mathbf{j}_{\mu-1}} \left(\mathbf{j}^{\mu-1} \cdot \mathrm{A} \cdot \mathbf{j}_\lambda - \sum_{\nu=0}^{\mu-1} \frac{\mathbf{j}^{\mu-1} \cdot \mathrm{A} \cdot \mathbf{j}_\nu \mathbf{j}^\nu \cdot \mathbf{j}_\lambda}{\mathbf{j}_\nu \cdot \mathbf{j}^\nu} \right) \tag{28}$$

Thus the vectors are biorthogonal in accordance with the equations

$$\mathbf{j}^\lambda \cdot \mathbf{j}_\mu = 0 \qquad\qquad \mathbf{j}^\mu \cdot \mathbf{j}_\lambda = 0 \tag{29}$$

for any $\lambda < \mu$ if $\mathbf{j}^\lambda \cdot \mathbf{j}_\nu = 0$ or $\mathbf{j}^\nu \cdot \mathbf{j}_\lambda = 0$ for $\nu \neq \lambda$. Since the vectors are biorthogonal for $\mu = 1$, they are biorthogonal for any μ by induction. The vectors are normalized in accordance with the equation

$$\mathbf{j}_\mu \cdot \mathbf{j}^\mu = 1 \tag{30}$$

Multiplication by $j^{\lambda-1} \cdot A$ gives the equation

$$j^{\lambda-1} \cdot A \cdot j_{\mu} = \frac{1}{j^{\mu} \cdot A \cdot j_{\mu-1}} \left(j^{\lambda-1} \cdot A \cdot A \cdot j_{\mu-1} - \sum_{\nu=0}^{\mu-1} j^{\lambda-1} \cdot A \cdot j_{\nu} j^{\nu} \cdot A \cdot j_{\mu-1} \right) \tag{31}$$

and multiplication by $A \cdot j_{\mu-1}$ gives the equation

$$j^{\lambda} \cdot A \cdot j_{\mu-1} = \frac{1}{j^{\lambda-1} \cdot A \cdot j_{\lambda}} \left(j^{\lambda-1} \cdot A \cdot A \cdot j_{\mu-1} - \sum_{\nu=0}^{\lambda-1} j^{\lambda-1} \cdot A \cdot j_{\nu} j^{\nu} \cdot A \cdot j_{\mu-1} \right) \tag{32}$$

Elimination of $j^{\lambda-1} \cdot A \cdot A \cdot j_{\mu-1}$ leads to the equation

$$\sum_{\nu=\lambda+1}^{\mu} (j^{\lambda-1} \cdot A \cdot j_{\nu})(j^{\nu} \cdot A \cdot j_{\mu-1}) = 0 \tag{33}$$

This equation can be true for all $\lambda \leqq \mu - 1$ only if the vectors satisfy the equation

$$j^{\lambda} \cdot A \cdot j_{\mu} = 0 \qquad (\lambda < \mu - 1) \tag{34}$$

Multiplication by $j^{\mu-1} \cdot A$ gives the equation

$$j^{\mu-1} \cdot A \cdot j_{\lambda} = \frac{1}{j^{\lambda} \cdot A \cdot j_{\lambda-1}} \left(j^{\mu-1} \cdot A \cdot A \cdot j_{\lambda-1} - \sum_{\nu=0}^{\lambda-1} j^{\mu-1} \cdot A \cdot j_{\nu} j^{\nu} \cdot A \cdot j_{\lambda-1} \right) \tag{35}$$

and multiplication by $A \cdot j_{\lambda-1}$ gives the equation

$$j^{\mu} \cdot A \cdot j_{\lambda-1} = \frac{1}{j^{\mu-1} \cdot A \cdot j_{\mu}} \left(j^{\mu-1} \cdot A \cdot A \cdot j_{\lambda-1} - \sum_{\nu=0}^{\mu-1} j^{\mu-1} \cdot A \cdot j_{\nu} j^{\nu} \cdot A \cdot j_{\lambda-1} \right) \tag{36}$$

Elimination of $j^{\mu-1} \cdot A \cdot A \cdot j_{\lambda-1}$ leads to the equation

$$\sum_{\nu=\lambda+1}^{\mu} (j^{\mu-1} \cdot A \cdot j_{\nu})(j^{\nu} \cdot A \cdot j_{\lambda-1}) = 0 \tag{37}$$

This equation can be true for all $\lambda \leqq \mu - 1$ only if the vectors satisfy the equation

$$j^{\mu} \cdot A \cdot j_{\lambda} = 0 \qquad (\lambda < \mu - 1) \tag{38}$$

Thus the vectors satisfy the three-term recurrence equations

$$j_{\mu} = \frac{1}{j^{\mu} \cdot A \cdot j_{\mu-1}} \left(A \cdot j_{\mu-1} - j_{\mu-1} j^{\mu-1} \cdot A \cdot j_{\mu-1} - j_{\mu-2} j^{\mu-2} \cdot A \cdot j_{\mu-1} \right) \tag{39}$$

and

$$j^{\mu} = \frac{1}{j^{\mu-1} \cdot A \cdot j_{\mu}} \left(j^{\mu-1} \cdot A - j^{\mu-1} \cdot A \cdot j_{\mu-1} j^{\mu-1} - j^{\mu-1} \cdot A \cdot j_{\mu-2} j^{\mu-2} \right) \tag{40}$$

Let $\beta_{\mu-1}$ be defined by the equation

$$\beta_{\mu-1} = j^{\mu-1} \cdot A \cdot j_{\mu-1} \tag{41}$$

and let $\gamma_{\mu-1}$ be defined by the equation

$$\gamma_{\mu-1} = j^{\mu} \cdot A \cdot j_{\mu-1} \equiv j^{\mu-1} \cdot A \cdot j_{\mu} \tag{42}$$

Then the three-term recurrence equations become

$$j_{\mu} = \frac{1}{\gamma_{\mu-1}} \left\{ A \cdot j_{\mu-1} - \beta_{\mu-1} j_{\mu-1} - \gamma_{\mu-2} j_{\mu-2} \right\} \tag{43}$$

and

$$j^{\mu} = \frac{1}{\gamma_{\mu-1}} \left\{ j^{\mu-1} \cdot A - \beta_{\mu-1} j^{\mu-1} - \gamma_{\mu-2} j^{\mu-2} \right\} \tag{44}$$

During each cycle of iteration the value of $\gamma_{\mu-1}$ is computed from the square root of the scalar product of the unnormalized vectors.

Insofar as the nth vectors j_n and j^n are orthogonal to n independent vectors, they must be zero within rounding error. Inasmuch as each vector is obtained from j_0 and j^0 by powers of the matrix A, the nth vectors must be the products of the vectors j_0 or j^0 and the characteristic polynomial $p(A)$ because by the Cayley–Hamilton theorem the matrix A is a solution of its own characteristic equation.

Let a sequence of polynomials $p_0(\alpha), \cdots, p_n(\alpha)$ be generated by a recurrence which starts with the equation

$$p_0(\alpha) = 1 \tag{45}$$

and continues with the equation

$$p_{\mu}(\alpha) = \frac{1}{\gamma_{\mu-1}} \left\{ \alpha p_{\mu-1}(\alpha) - \beta_{\mu-1} p_{\mu-1}(\alpha) - \gamma_{\mu-2} p_{\mu-2}(\alpha) \right\} \tag{46}$$

until $p_n(\alpha) = p(\alpha)$. The nth polynomial and its derivatives can be utilized by a general routine for the determination of the real and complex roots of the characteristic polynomial.

Insofar as the matrix A is expressed by the equation

$$A = \sum_{\nu=1}^{n} \alpha_{\nu} a_{\nu} a^{\nu} \tag{47}$$

a polynomial $p_{\mu}(A)$ is expressed by the equation

$$p_{\mu}(A) = \sum_{\nu=1}^{n} p_{\mu}(\alpha_{\nu}) a_{\nu} a^{\nu} \tag{48}$$

The vectors j_{μ} and j^{μ} are expressed by the equations

$$j_{\mu} = p_{\mu}(A) \cdot j_0 \qquad\qquad j^{\mu} = j^0 \cdot p_{\mu}(A) \tag{49}$$

Inasmuch as the vectors j_{μ} and j^{μ} are biorthogonal, they can be used in an expansion of the identity matrix I in accordance with the equations

$$I = \sum_{\nu=0}^{n-1} j_{\nu} j^{\nu} \qquad\qquad I = \sum_{\nu=0}^{n-1} j^{\nu} j_{\nu} \tag{50}$$

Multiplication of the characteristic vectors by the identity matrix and substitution for vectors their expressions in terms of polynomials lead to the equations

$$a_{\mu} = (j^0 \cdot a_{\mu}) \sum_{\nu=1}^{n} p_{\nu}(\alpha_{\mu}) j_{\nu} \tag{51}$$

and

$$a^{\mu} = (j_0 \cdot a^{\mu}) \sum_{\nu=1}^{n} p_{\nu}(\alpha_{\mu}) j^{\nu} \tag{52}$$

The accuracy of the roots and vectors is limited by the extent to which the initial vectors have substantial components in the directions of all of the characteristic vectors of the matrix.

If the matrix has a multiple root, it is necessary to repeat the iteration with new initial vectors until all of the vectors have been determined for the multiple root.

APPENDIX C

TEST MATRICES

CATALOG OF MATRICES

Test Matrices for $n = 6$ in Card Decks

Let A = identity matrix.

$$a_{ij} = 0 \qquad (i \neq j)$$

$$a_{ij} = 1 \qquad (i = j)$$

$A_1 = A = I = A^{-1}$

Let A = circulative permutation.

$$a_{ij} = 0 \qquad (j \neq (i + 1)\text{modulo}\, n)$$

$$a_{ij} = 1 \qquad (j = (i + 1)\text{modulo}\, n)$$

$A_2 = A \qquad A^{-1} = A'$

Let A = reversive permutation.

$$a_{ij} = 0 \qquad (j \neq n + 1 - i)$$

$$a_{ij} = 1 \qquad (j = n + 1 - i)$$

$A_3 = A = A' = A^{-1}$

Let A = asymmetric permutation.

$$a_{ij} = 0 \qquad (j \neq (n - 2i + 1)\text{modulo}(n + 1))$$

$$a_{ij} = 1 \qquad (j = (n - 2i + 1)\text{modulo}(n + 1))$$

$A_4 = A$

Let A = positive definite circulative matrix.

$$a_{ij} = 1 \qquad (j \neq i)$$

$$a_{ij} = n + 1 \qquad (j = i)$$

$A_5 = A$

Let A = inverse circulative matrix.

$$a_{ij} = -\frac{1}{2n^2} \qquad (j \neq i)$$

$$a_{ij} = +\frac{2n-1}{2n^2} \qquad (j = i)$$

$A_6 = (72)A = (72)A_5^{-1}$

Let A = tridiagonal finite difference matrix.

$$a_{ij} = 0 \qquad (|j - i| > 1)$$

$$a_{ij} = -1 \qquad (|j - i| = 1)$$

$$a_{ij} = +2 \qquad (|j - i| = 0)$$

$A_7 = A$

Let A = inverse tridiagonal matrix.

$$a_{ij} = \frac{n+1-j}{n+1}\,i \qquad (j \geqq i)$$

$$a_{ij} = \frac{n+1-i}{n+1}\,j \qquad (i \geqq j)$$

$A_8 = (7)A = (7)A_7^{-1}$

Let A = symmetric Frank matrix.

$$a_{ij} = \min(i, j)$$

$A_9 = A$

Let A = inverse Frank matrix.

$$a_{ij} = 0 \qquad (|j - i| > 1)$$

$$a_{ij} = -1 \qquad (|j - i| = 1)$$

$$a_{ij} = +2 \qquad (i = j \neq n)$$

$$a_{ij} = +1 \qquad (i = j = n)$$

$A_{10} = A = A_9^{-1}$

Let **A** symmetric Boothroyd matrix.

$$a_{ij} = \max(i, j)$$

$\mathbf{A}_{11} = \mathbf{A}$

Let **A** = inverse Boothroyd matrix.

$$\begin{array}{ll} a_{ij} = 0 & (|j - i| > 1) \\ a_{ij} = +1 & (|j - i| = 1) \\ a_{ij} = -1 & (i = j = 1) \\ a_{ij} = +2 & (i = j,\ 1 < i < n) \\ a_{ij} = -\dfrac{n-1}{n} & (i = j = n) \end{array}$$

$\mathbf{A}_{12}$ $(6)\mathbf{A} = (6)\mathbf{A}_{11}^{-1}$

Let **A** symmetric Givens matrix.

$$a_{ij} = 2\min(i, j) - 1$$

$\mathbf{A}_{13}$ $\mathbf{A}$

Let **A** inverse Givens matrix.

$$\begin{array}{ll} a_{ij} = 0 & (|j - i| > 1) \\ a_{ij} = -\frac{1}{2} & (|j - i| = 1) \\ a_{ij} = +\frac{3}{2} & (i = j = 1) \\ a_{ij} = +1 & (i = j,\ 1 < i < n) \\ a_{ij} = +\frac{1}{2} & (i = j = n) \end{array}$$

$\mathbf{A}_{14}$ $(2)\mathbf{A}$ $(2)\mathbf{A}_{13}^{-1}$

Let A = symmetric Lehmer matrix.

$$a_{ij} = \frac{i}{j} \qquad (i \leqq j)$$

$$a_{ij} = \frac{j}{i} \qquad (j \leqq i)$$

$A_{15} = (60)A$

Let A = inverse Lehmer matrix.

$$a_{ij} = 0 \qquad (|j - i| > 1)$$

$$a_{ij} = -\frac{i + 1}{2i + 1}\, i \qquad (j = i + 1)$$

$$a_{ij} = -\frac{j + 1}{2j + 1}\, j \qquad (j = i - 1)$$

$$a_{ij} = +\frac{4i^3}{4i^2 - 1} \qquad (i = j,\ i < n)$$

$$a_{ij} = +\frac{n^2}{2n - 1} \qquad (i = j = n)$$

$A_{16} = (3465)A = (207900)A_{15}^{-1}$

Let A = symmetric Todd matrix.

$$a_{ij} = |i - j|$$

$A_{17} = A$

Let A = inverse Todd matrix.

$$a_{ij} = 0 \qquad (1 < |j - i| < n - 1)$$

$$a_{ij} = +\tfrac{1}{2} \qquad (|j - i| = 1)$$

$$a_{ij} = +\frac{\frac{1}{2}}{n - 1} \qquad (|j - i| = n - 1)$$

$$a_{ij} = -\frac{n - 2}{2n - 2} \qquad (i = j = 1)$$

$$a_{ij} = -1 \qquad (i = j,\ 1 < i < n)$$

$$a_{ij} = -\frac{n - 2}{2n - 2} \qquad (i = j = n)$$

$A_{18} = (10)A = (10)A_{17}^{-1}$

Let A = symmetric Lietzke matrix

$$a_{ij} = n - |i - j|$$

$A_{19} = A$

Let A = inverse Lietzke matrix.

$$a_{ij} = 0 \qquad (1 < |j - i| < n - 1)$$

$$a_{ij} = -\tfrac{1}{2} \qquad (|j - i| = 1)$$

$$a_{ij} = +\frac{1}{2n + 2} \qquad (|j - i| = n - 1)$$

$$a_{ij} = +\frac{n + 2}{2n + 2} \qquad (i = j = 1)$$

$$a_{ij} = +1 \qquad (i = j,\ 1 < i < n)$$

$$a_{ij} = +\frac{n + 2}{2n + 2} \qquad (i = j = n)$$

$A_{20} = (14)A^{-1} = (14)A_{19}^{-1}$

Let A = symmetric Pascal matrix.

$$a_{ij} = \frac{(i + j - 2)!}{(i - 1)!(j - 1)!}$$

$A_{21} = A$ $\qquad\qquad$ $A_{22} = A^{-1}$

Let A = trigonometric orthogonal matrix.

$$a_{ij} = \left(\frac{2}{n + 1}\right)^{\frac{1}{2}} \sin\left(\frac{ij\pi}{n + 1}\right)$$

$A_{23} = A = A' = A^{-1}$

Let A = Hilbert matrix.

$$a_{ij} = \frac{1}{i+j-1}$$

$A_{24} = (27720)A$

Let A = inverse Hilbert matrix.

$$a_{ij} = \frac{(-1)^{i+j}(n+i-1)!(n+j-1)!}{(i+j-1)(i-1)!^2(j-1)!^2(n-i)!(n-j)!}$$

$A_{25} = A = (27720)A_{24}^{-1}$

Let A = symmetric Hadamard matrix.

Let k be any integer and let $p = n + 1$ where p is a prime number. The Legendre–Jacobi reciprocity function,

$$\left(\frac{k}{p}\right) \equiv \begin{cases} -1 \text{ if } k \neq mp, k \neq m^2 \text{modulo}\, p \\ \;\;0 \text{ if } k = mp \\ +1 \text{ if } k = m^2 \text{modulo}\, p \end{cases}$$

where m is any integer. Then

$$a_{ij} = \left(\frac{i+j}{n+1}\right)$$

$A_{26} = A$

Let A = inverse Hadamard matrix.

$$a_{ij} = \frac{1}{n+1}\left[\left(\frac{i+j}{n+1}\right) - \left(\frac{i}{n+1}\right) - \left(\frac{j}{n+1}\right)\right]$$

$A_{27} = (7)A = (7)A_{26}^{-1}$

Let A = random Forsythe matrix.

$$a_{ij} = \text{random number } m \text{ in range } -100 < m < +100$$

$A_{28} = A$

Let A = Clement matrix.

$$\begin{array}{ll} a_{ij} = 0 & (|j - i| \neq 1) \\ a_{ij} = i & (j = i + 1) \\ a_{ij} = n - j & (j = i - 1) \end{array}$$

$A_{29} = A$ $A_{30} = (15)A_{29}^{-1}$

Let A = linear circulative matrix.

$$a_{ij} = (j - i + 1)\text{modulo } n$$

$A_{31} = A$

Let A = inverse circulative matrix.

$$\begin{array}{ll} a_{ij} = +\dfrac{2}{n^2(n+1)} & (i \neq j, j \neq (i+1)\text{modulo } n) \\ a_{ij} = +\dfrac{n^2+n+2}{n^2(n+1)} & (j = (i+1)\text{modulo } n) \\ a_{ij} = -\dfrac{(n-1)(n+2)}{n^2(n+1)} & (i = j) \end{array}$$

$A_{32} = (126)A = (126)A_{31}^{-1}$

Let A = permuted Wilkinson matrix.

$$\begin{array}{ll} a_{ij} = (-1)^{i+j} & (j \leqq i) \\ a_{ij} = 1 & (j = i + 1) \\ a_{ij} = 0 & (j > i + 1) \end{array}$$

$A_{33} = A$ $A_{34} = A^{-1}$

Let A = binomial matrix

$$a_{ij} = \frac{(-1)^{j-1}(i-1)!}{(i-j)!(j-1)!}$$

$A_{35} = A + A^{-1}$

Let A = Vandermonde matrix.

$$a_{ij} = j^{i-1}$$

$A_{36} = A$ $\qquad\qquad$ $A_{37} = (120)A_{36}^{-1}$

Let A = interpolative matrix.

$$a_{ij} = \left(\frac{i-1}{n-1}\right)^{j-1}$$

$A_{38} = A$ $\qquad\qquad$ $A_{39} = (24)A_{38}^{-1}$

Let A = nondefective companion matrix.

$$\begin{array}{ll} a_{ij} = 0 & (i \neq n, j \neq i+1) \\ a_{ij} = +1 & (j = i+1) \\ a_{ij} = -1 & (i = n) \end{array}$$

$A_{40} = A$

Let A = special symmetric Martin matrix.

This matrix has three pairs of equal roots.

$A_{41} = A$

Let A = special symmetric Voigt matrix.

This matrix has three pairs of equal roots.

$A_{42} = A$

Let A = special nonsymmetric Varah matrix.

The roots are $(-3, -2, \frac{1}{3}, 1, \frac{3}{2}, 2)$ for which there are six independent pairs of vectors.

A_{43} = (6)A

Let A = defective companion matrix.

$$a_{ij} = 0 \qquad (i \neq n, j \neq i + 1)$$

$$a_{ij} = 1 \qquad (j = i + 1)$$

$$a_{ij} = \frac{(-1)^{n-j} n!}{(n - j)! j!} \qquad (i = n)$$

This matrix has six roots equal to 1, for which there is only one pair of vectors.

A_{44} = A $\qquad$ A_{45} = A^{-1}

Let A = defective triangular matrix.

$$a_{ij} = (-1)^{i-j} \qquad (i \geqq j)$$

$$a_{ij} = 0 \qquad (j > i)$$

This matrix has six roots equal to 1, for which there is only one pair of vectors.

A_{46} = A $\qquad$ A_{47} = A^{-1}

Let A = special defective Voigt matrix.

The roots are $(-1, -1, -1, -i, +i, +1)$ for which the three roots equal to -1 have only one pair of vectors.

A_{48} = A

Let A = special defective Varah matrix.

The roots are $(1, 1, 2 - i, 2 + i, 3, 3)$ for which the two roots equal to $+1$ have only one pair of vectors.

A_{49} = A

Let A = symmetric singular matrix.

The roots are (0, 0, 3, 4, 4, 13) for which there are six independent pairs of vectors.

A_{50} = A

APPENDIX D

DISTRIBUTION

DISTRIBUTION

Defense Documentation Center
Cameron Station
Alexandria, VA 22314 (12)

Defense Printing Service
Washington Navy Yard
Washington, DC 20374

Library of Congress
Washington, DC 20540
Attn: Gift and Exchange Division (4)

C
D
E
F
G
K
K05H
K05S
K05D
K10
K20
K30
K40
K50
K60
K70
K80
R
U
V
X21 (2)
X220 (6)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER
NSWC/DL TR-3689 ✓

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
MATRIX ARITHMETIC AND CHARACTERISTICS COMPUTATION

5. TYPE OF REPORT & PERIOD COVERED

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
A. V. Hershey

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS ✓
Naval Surface Weapons Center (K05)
Dahlgren, Virginia 22448

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
NIF

11. CONTROLLING OFFICE NAME AND ADDRESS

12. REPORT DATE
October 1977

13. NUMBER OF PAGES
50

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

matrices
inversion
characteristics
pivoting
partitioning
analysis
programming
computation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Analysis and documentation are given for subroutines which do matrix arithmetic and characteristics computation. The subroutines make it possible for the elements of the matrix to be in the natural arrangement. During inversion or trianguloidization the pivot selection is independent of the scaling of the rows and columns. A subroutine does arithmetic on partitioned matrices.

DA

FIL

4 -